REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	3 August 1999	Final Report

4. TITLE AND SUBTITLE

Intelligent Process Control Via Gaze Detection Technology

5. FUNDING NUMBERS

F6256299M9090

6. AUTHOR(S)

Prof. Jaihie Kim

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Yonsei University
Department of Electronic Engineering
Sudaemoon-Gu
Seoul 120-749
Korea (South)

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AOARD
UNIT 45002
APO AP 96337-5002

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AOARD98-09

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

A

13. ABSTRACT (Maximum 200 words)

Gaze detection is to locate the position on a monitor screen where a user is looking. In their work, they implemented it with a computer vision system setting a camera above a monitor, and a user moves (rotates and/or translates) their face to gaze at a different position on the monitor. For their case, the user was requested not to move pupils of their eyes when they gazed at a different position on the monitor screen, though they are working on relaxing this restriction. Up to now, they have tried several different methods and among them the proposed Two Neural Network Method shows the best results. For the application of the gaze detection techniques to the user-interface of a control program, they supplemented two additional techniques. One was to drag a mouse cursor by moving their face after initially placing the cursor by gazing, and another was to click the cursor by winking one eye. These were applied to user-interface of a process control program in three different ways. First, even when both hands are busy doing the work, the user can still control the monitor screen by gazing, dragging, and winking. Secondly, if the user is absent and not in front of the monitor when important states occur, then the system can record and replay them later when the user returns. Finally, as the system knows where the user is looking, if an emergency occurs at a different point from where the user is looking, then the system can display a warning signal. This paper also includes subjective and objective experimental results obtained from the tests by applying the underlined techniques to a process control for Chemical Vapor Decomposition.

14. SUBJECT TERMS

Gaze Detection Technology, Computer Vision System, Two Neutral Network Method

15. NUMBER OF PAGES
12

16. PRICE CODE
N/A

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19, SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18
298-102

INTELLIGENT PROCESS CONTROL VIA GAZE DETECTION TECHNOLOGY

ABSTRACT

Gaze detection is to locate the position on a monitor screen where a user is looking. In our work, we implement it with a computer vision system setting a camera above a monitor, and a user moves (rotates and/or translates) her face to gaze at a different position on the monitor. For our case, the user is requested not to move pupils of her eyes when she gazes at a different position on the monitor screen, though we are working on to relax this restriction. Up to now, we have tried several different methods and among them the proposed Two Neural Network Method shows the best results. For the application of the gaze detection techniques to the user-interface of a control program, we supplement two additional techniques. One is to drag a mouse cursor by moving her face after initially placing the cursor by gazing, and another is to click the cursor by winking one eye. These are applied to user-interface of a process control program in three different ways. Firstly, even when both hands are busy in doing a work, the user can still control the monitor screen by gazing, dragging and winking. Secondly, if the user is absent and not in front of the monitor when important states occur, then system can record and replay them later when the user returns. Finally, as the system knows where the user is looking, if an emergency occurs at a different point from where user is looking, then the system can display a warning signal. This paper also includes subjective and objective experimental results obtained from the tests by applying the underlined techniques to a process control for Chemical Vapor Decomposition.

1. INTRODUCTION

Gaze detection is to locate the position on a monitor screen where a user is looking, and it is applicable to many areas that users cannot use their hands for communicating with computers. These include view controls in three dimensional simulation programs, man-machine interfaces for handicapped persons and intelligent interfaces for process control systems, etc. In our work, we implement it with a computer vision system setting a camera above the monitor screen, and the user moves (rotates and translates) his(her) face to gaze at different positions on the monitor. Here, the translation means a movement without any rotation. At this moment, the user is requested not to move the pupils of her eyes when she gazes at a different position on the monitor screen. We are currently working to relax this restriction. This way of detecting the gazing position on a monitor screen by computer vision is a recently developing area[1], although computing the face movement or gazing vector alone has been studied by some researchers[2][3][4][5]. Rikert et al.[1] locate the gaze point on a monitor using morphable models which are trained by neural networks, but their method has a constraint that the distance of the user from the monitor must be fixed for all training and testing periods, which severely restricts practical usage. In our approach, the distance between a user and a monitor is normalized by the feature values obtained from a user's face for our system to be independent of the distance, and two separate neural networks for computing the gaze position for facial rotation and translation are devised allowing a moderate amount of translation in addition to rotation.

In our approach, gaze detection proceeds as follows. From the image obtained through a camera, the user's face is located. Then, features including two eyes, nostrils and lip corners within the face region are extracted. Useful feature values are obtained from the positions of those features and the geometrical shapes they are forming in the image. These feature values are compared to those values when the user looks at the center of monitor. The differences of those feature values are inputted to the network computing face rotation and the difference between face positions is inputted to the network computing face translation. Each of these two networks is trained separately by allowing only rotation and translation, respectively. Together, they detect a relatively correct gaze position even when a moderate amount of face translation has occurred.

A user can place a mouse cursor or a prompt on the monitor by gazing at her desired position. There may be system's detection error, but the user can drag the cursor by moving her face again toward the desired position. When the cursor stops at the position, the user can wink her eye for a clicking effect. Thus, this way of gazing, dragging and winking can function together like an ordinary computer mouse for a man-computer interface.

In this paper, the gaze detection technology is applied to user-interfaces of a process control program in three different ways. Firstly, even when both hands are busy in controlling a complicated monitor panel or recording some important matters, the user can still control the monitor by gazing, dragging and winking. Secondly, if the user is absent and not in front

of the monitor when some important states occur, then the system records and replays them later when the user returns. Finally, as the system knows where the user is looking, if an emergency occurs at a different position that the user is not looking, then system can display a warning message at the gaze area.

2. Techniques for Gaze Detection

2.1 Extraction of a Face Region and Facial Features

Locating a user's face and facial features in an input image is the first step for gaze detection. To compute the amount of face rotation, facial features are used. When a user starts to use the system, facial features are extracted in the face region after initially locating the face region in the image. Once the facial features are located, there is no need to search the face region again in the next image, for the feature positions are tracked by predicting their next positions from their current positions.

From images captured by camera, a face region is detected by difference images. Difference images are usually enough to estimate the locations of face regions in images, for even a person thinks she does not move her face, a typical difference image notifies even a little movement of her face. However, this way of detecting face regions may not be accurate enough for the purpose of computing a face movement, as discussed in the next section. After a face region in an image is found, typical locations of facial features like both eyes, nostrils and lip corners are predicted. Within the predicted regions, accurate position of each feature is identified by vertical and horizontal histogram methods[7]. Table 1 shows our experimental results in average RMS errors obtained by this method.

(unit: pixel)

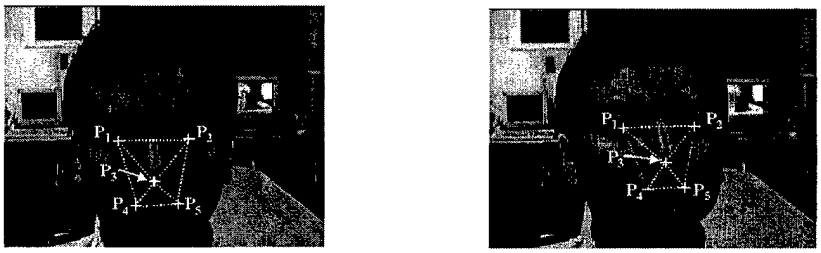
Feature point	Eyes	Nostrils	Lip corners
RMS error	2	5	3

Table 1: RMS error between detected feature positions and real feature positions

Average sizes of an entire input image, a face, an eye, a nose and a lip corner are 320*240, 208*144, 8*4, 4*4 and 3*3 pixels, respectively. We test 2000 successive images (100 images*20 persons) under the assumption that the surrounding brightness is not changed much.

2.2 Gaze Detection by Neural Networks

Fig.1 (a) and Fig.1.(b) show the facial feature points and some geometrical shapes composed of them when a user is looking at the center of the monitor screen and another point on the monitor, respectively.



(a) Gazing at the center of a monitor (b) Gazing at another point on a monitor

Fig. 1. Feature positions when a user gazes at monitor center and another point on a monitor

Differences of these feature positions and geometrical shapes composed of them are inputted to the neural networks for gaze detection. The number of inputs is twenty, and they are obtained as follows.

I When a user gazes at monitor center

P_1 (left eye : X_1, Y_1), P_2 (right eye : X_2, Y_2), P_3 (nose : X_3, Y_3), P_4 (left lip corner : X_4, Y_4), P_5 (right lip corner : X_5, Y_5)

I When a user gazes at another point on a monitor

P'_1(left eye : X_1, Y_1), P'_2(right eye : X_2, Y_2), P'_3(nose : X_3, Y_3), P'_4(left lip corner : X_4, Y_4), P'_5(right lip corner : X_5, Y_5)

FV(Feature Value)$_{1\sim5}$: X '$_i$ - X$_i$, FV$_{6\sim10}$: Y '$_i$ - Y$_i$ (i = 1, ... 5)

FV$_{11}$: S (P'$_1$ P'$_2$ P'$_3$) - S(P$_1$ P$_2$ P$_3$) FV$_{12}$: S (P'$_1$ P'$_3$ P'$_4$) - S(P$_1$ P$_3$ P$_4$)

FV$_{13}$: S (P'$_2$ P'$_3$ P'$_5$) - S(P$_2$ P$_3$ P$_5$) FV$_{14}$: S (P'$_3$ P'$_4$ P'$_5$) - S(P$_3$ P$_4$ P$_5$)

FV$_{15}$: S (P'$_1$ P'$_3$ P'$_4$) / S(P'$_2$ P'$_3$ P'$_5$) - S (P$_1$ P$_3$ P$_4$) / S(P$_2$ P$_3$ P$_5$)

FV$_{16}$: S (P'$_3$ P'$_4$ P'$_5$) / S(P'$_1$ P'$_2$ P'$_3$) - S (P$_3$ P$_4$ P$_5$) / S(P$_1$ P$_2$ P$_3$)

FV$_{17}$: {(X'$_1$+X'$_4$)/2-X'$_3$}-{(X$_1$+X$_4$)/2-X$_3$}, FV$_{18}$: {X'$_3$-(X'$_2$+X'$_5$)/2}-{X$_3$-(X$_2$+X$_5$)/2}

FV$_{19}$: {(Y'$_4$+Y'$_5$)/2-Y'$_3$}-{(Y$_4$+Y$_5$)/2-Y$_3$}, FV$_{20}$: {Y'$_3$-(Y'$_1$+Y'$_2$)/2}-{Y$_3$-(Y$_1$+Y$_2$)/2}

where, S (P'$_1$P'$_2$P'$_3$) is the triangle size determined from P'$_1$P'$_2$P'$_3$.

Sitting heights of training users and their distances from the monitor may be different from those of testing users, which could be one cause of the gaze detection error. In our case, the camera is set above the monitor for the sitting heights of various users to be less changeable, but we still need to normalize the distances of users from the monitor. For this normalization, a user is asked to look at the predefined right-most upper point, the left-most lower point and the center point on the monitor before her gazing point is detected. We obtain the twenty feature values by this way and use them for normalization.

Gaze Detection by Single Neural Network

The first method we tried for gaze detection was using one neural network for computing facial rotation only. This multi-layer Perceptron accepts twenty feature values as inputs and outputs the X and Y coordinate points as the detected gaze position on the monitor. For the output functions of the neural net, linear function, sigmoid function and inverse sigmoid function are tested to find that the linear function shows the best performance.

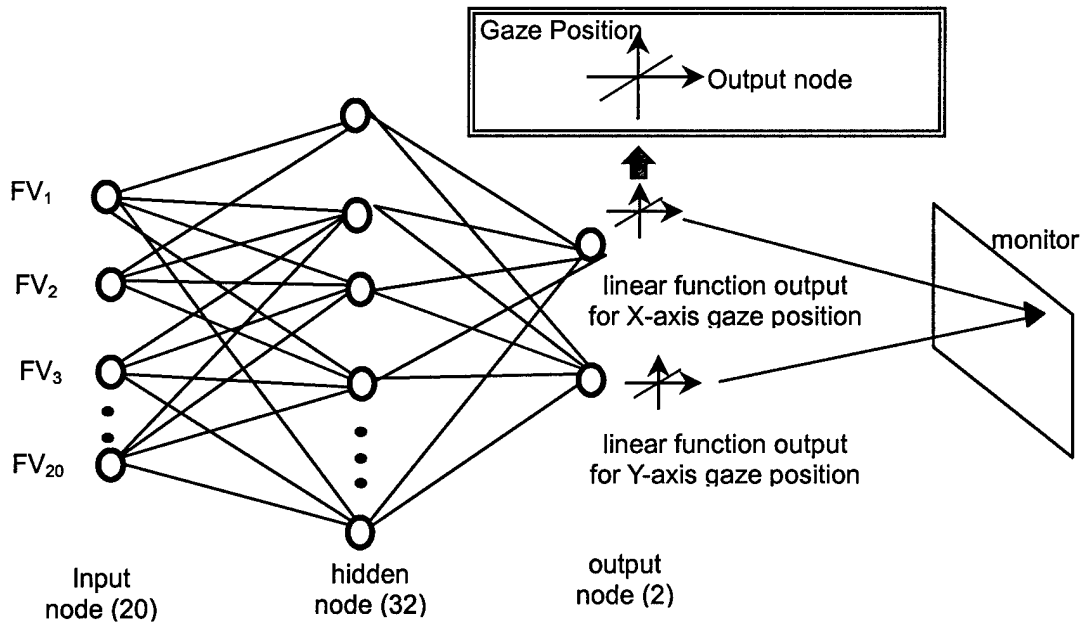


Fig. 2. Single Neural Network for Computing Gaze Detection

The generalized delta rule as shown in equation (1) was used to train the neural net iteratively from 10000 up to 50000 times

Gaze Detection by Two Neural Networks

$$w_{kj}(t+1) = w_{kj}(t) + \eta \delta_{pk} i_{pk} \tag{1}$$

$w_{kj}(t)$: weight on each node at t iterations

δ_{pk} : error terms of hidden units or output units

η : learning rate parameter (=0.01)

i_{pj} : output value calculated from input layer or hidden layer

The twenty feature values are effective for gaze detection when user's face makes rotation only. However, face translation needs be considered separately and differently. Thus, we propose Two Neural Networks Method to compute the amounts of face rotation and translation separately. Estimation of a face position by a difference image is useful for obtaining facial features, but the outline of a face obtained from it is blurred, noisy and sometimes not enough for computing the accurate face position. For a better computation of a face translation, the positions or outline of a face is located by a template matching. The template for locating a face contains a generalized shape of face outlines, and the size of face in template is adaptive and normalized by the distance between two eyes of the user. Thus it becomes independent of the face size of a user and the distance between a user and a monitor. Positional differences of templates obtained from the images when the user is looking at the center and another position on the monitor are entered as X and Y inputs to the neural network for computing face translation. This is shown in Fig.3. Outputs from two separate networks are merged to give a final gaze position.

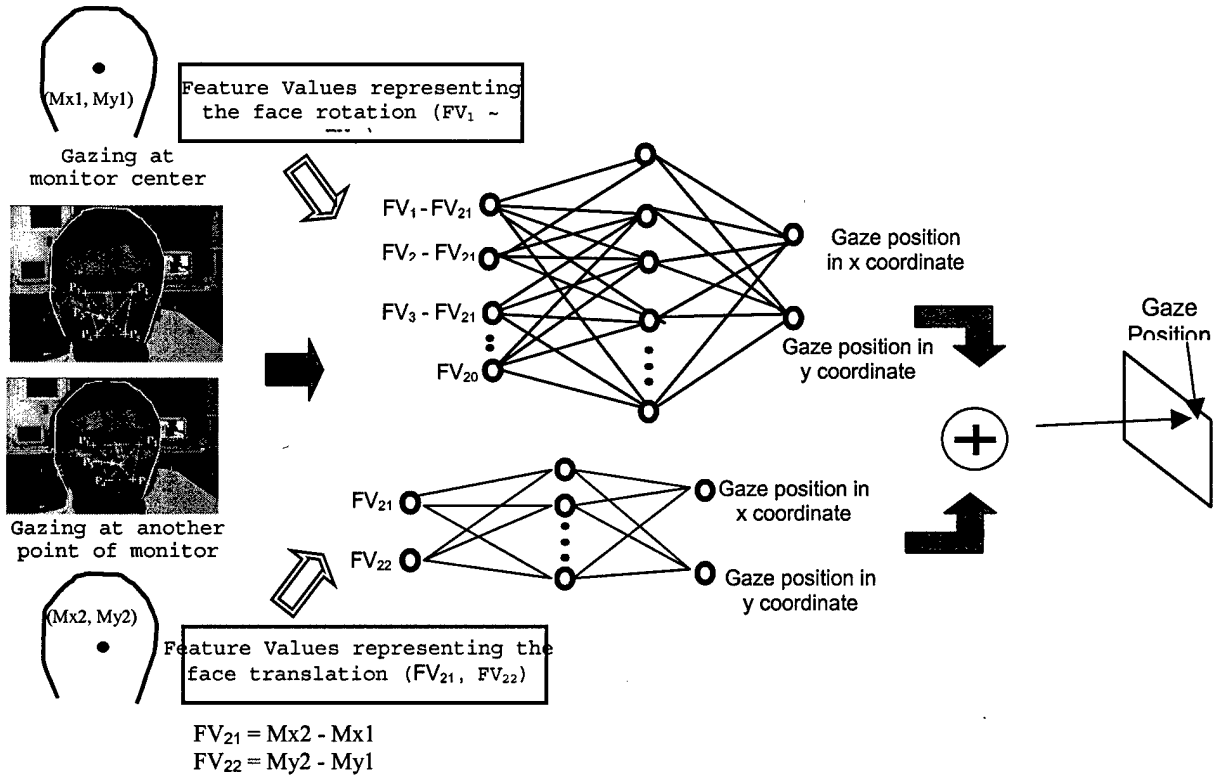


Fig. 3. Gaze detection by two neural networks

For the training purpose, ten users were asked to gaze at twenty-two predefined positions on a 19-inch monitor. The testing data were obtained in the same way as the training data, but they were not used for training. Those users were sitting from 50 to 70 cm in front of the monitor. Table 2 shows experimental results obtained when faces make rotations only.

(unit: inch)

Method	Linear Interpolation	**Single Neural Net**	Two Neural Nets
RMS error	1.84	**1.64**	1.7

Table 2: Gaze detection errors when faces make rotation only (5 persons * 22 gaze positions = 110 test data).

In Table 2, the Linear Interpolation Method[6] is a simple method where training data describing the input-output values are stored, and for new-coming inputs their outputs are generated by interpolating the outputs from the stored data near to them. When only rotations are occurred, the Single Neural Network Method having only one network for computing rotations

shows slightly better results than the Two Neural Networks Method or the Linear Interpolation Method. However, when both rotations and a moderate amount of translations are occurred, the Two Neural Networks Method shows better results than the other two methods as shown in Table 3.

(unit: inch)

Method	Linear Interpolation	Single Neural Net	**Two Neural Nets**
RMS error	4.54	4.40	**3.4**

Table 3: Gaze detection error by face rotation and translation (5 persons * 22 gaze positions = 110 data)

The RMS error resulting from the Two Neural Networks method, 3.4 inch, is smaller than the other two methods, but still too much for some cases. However, dragging the cursor as described in Section 3 may compensate the detection error. The error is caused by the following reasons.

1) To detect the position of a face, a simple template matching using a generalized shape of face outlines is used. This is very fast, but it is not robust enough so that sometimes it fails to locate the exact position of a face as shown in Figure 4. The average RMS error is 29 pixels (22 in X-axis and 19 in Y-axis), which becomes a major source of error in computing the amount of face translation.
2) As shown in Table 1, errors in detecting facial features are also causes of final gaze detection error.
3) Although users are requested not to move their pupils of eyes, they unintentionally move them somewhat. It becomes another source of the gaze detection error.

Fig. 4. The examples of location error of face

Currently, we are working to reduce these sources of detection error. We have also considered a gaze detection method computing in a three dimensional world; namely, gaze points on a monitor are considered as the intersecting points between the monitor screen and the normal vector to the plane formed by facial feature points. This method requires complex computations requiring pre-defined geometrical and camera parameters. It will be reported later in another paper. So far, the Two Neural Networks Method seems to be the best as far as the processing times and our applications are concerned.

3. Additional Techniques for Practical Use

For applying the gaze detection techniques to user-interfaces of control programs, we supplement two additional techniques. One is to compensate the gaze detection error. Namely, by moving user's face toward the desired point on the monitor, the monitor cursor initially placed by gazing is dragged to the point. The cursor stops by halting the face movement. This way of dragging a cursor by moving face can be accomplished by tracing the positions of facial features in images. Another technique is to offer the function corresponding to clicking a mouse button. This is to send a command signal to the system when the cursor is placed at the desired position. This function is accomplished by winking one eye. Winking one eye can be detected by comparing the sizes and the ratios of width to height of two eyes as shown in Figure 5.

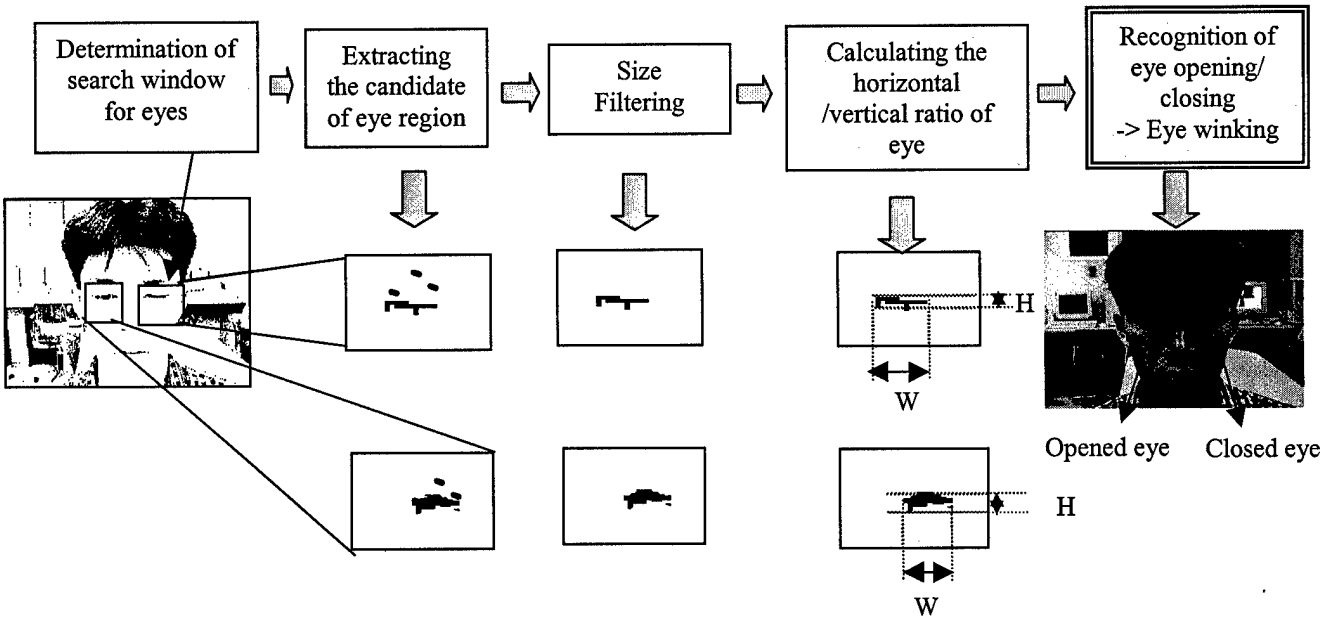


Fig .5. Recognition of Winking Eye

4. Application of Gaze Detection to a Process Control

4.1 Three Functions for a Chemical Vapor Deposition Process

CVD (Chemical Vapor Deposition) is a process utilized in many areas including fabrication and surface coating of semiconductor devices. Fig. 6 shows a typical example of a CVD process control. The gaze detection is used to support the following three functions.

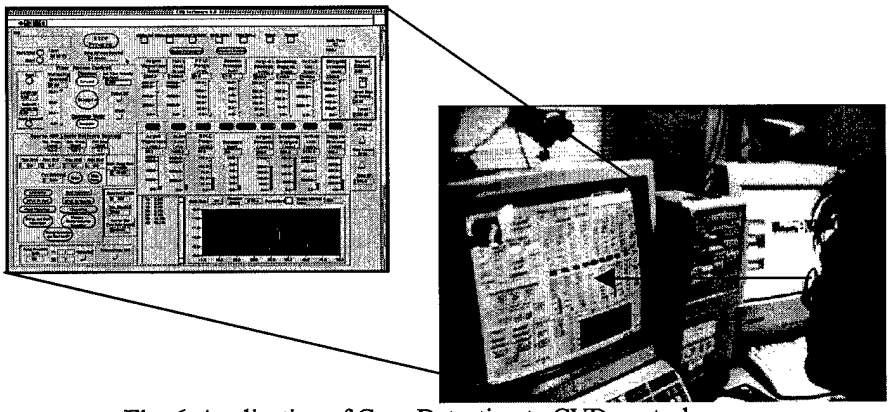


Fig. 6. Application of Gaze Detection to CVD control program.

Function 1 : Process control when both hands are busy

In a CVD process, a user may need to control the process even when both hands are busy in doing other things, as shown in Fig.7.

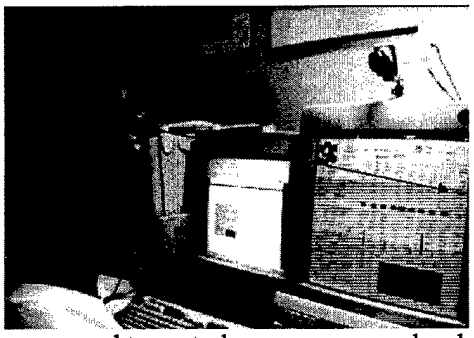


Fig. 7. A user may need to control a process even when both hands are busy.

For example, as shown in Fig.8, the pipes (a), (b) and (c) carrying oxygen, argon and vaporizer need be opened at the same time, or if the pump is not working well, then the pipes (1), (2) and (3) need to be closed at the same time. Or, sometimes a user may want to control a switch on monitor while she is recording on a process.

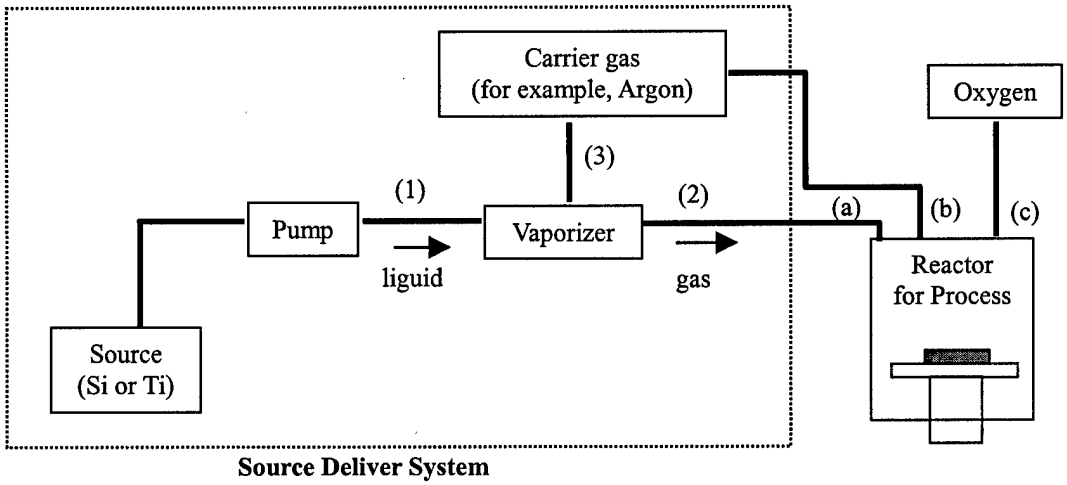


Fig. 8.A CVD control system

Function 2 : Replay when the user returns

As we always trace the facial features, if these features are disappeared, then we may suspect the user is not present. Thus it is possible to record automatically some important data (for examples, pressures and temperatures in the reactor) while the user is absent and replay them later when the user returns.

Function 3 : Leads a user's attraction to important spots

If an emergent signal or a state appears on the monitor when the user is looking at some other place, the system is able to find that the user is not looking at the right place and lead the user's attraction to it. This is useful when gas leaks or the temperature and/or pressure in the reactor increase too much. Namely, if an emergency occurs at the pump where the user is not looking, then the lamp can be twinkling as a warning signal. Or, the system can display a warning message at the place where user is currently looking.

4.2 State Transitions for User-Interface

Fig.9 shows the state transition diagram to implement the three functions for the CVD control.

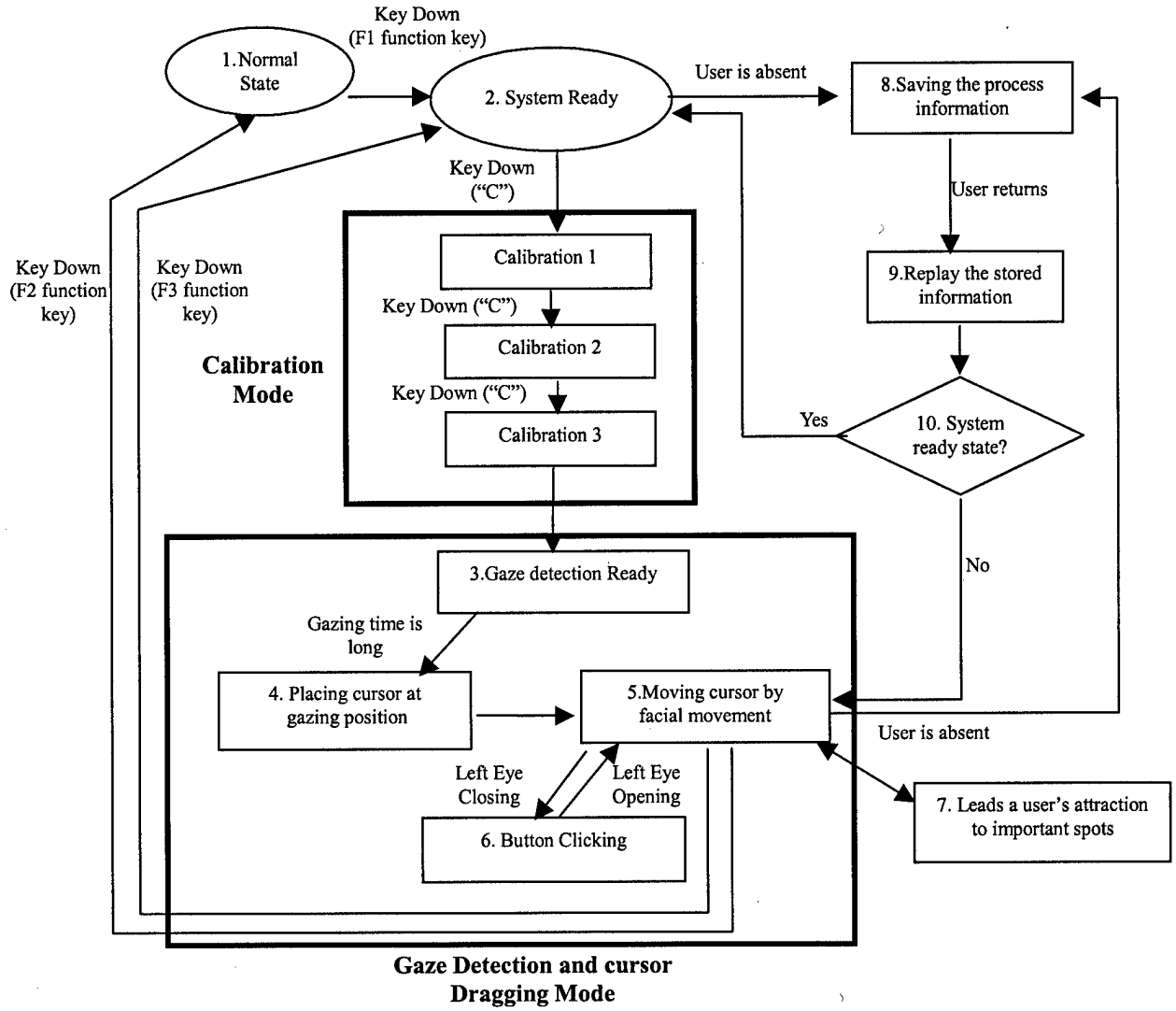


Fig. 9. State Transition Diagram for CVD Control.

State 1 is for the normal computer mode where the gaze detection is not used. By pressing the special function keys (F1), the system enters into the State 2, the ready mode, where the user's facial features are tracked. Pressing the key 'C' in State 2, the system enters into the calibration mode where the user is asked to gaze at three predefined points on the monitor. This is for the normalization purpose as described in Section 2. When calibrations are over, the system automatically enters into State 3 where gaze detection is ready. In this state, if the user gazes at a certain point for more than 'predefined threshold' seconds, then the gazed position is detected and a cursor or prompt is placed at the detected point (State 4). Then system enters into State 5 by itself, where the cursor is dragged by face movement. If the user want to send a control signal like clicking the mouse button, she can do it by winking her left eye. In the state 2 or 5, if facial features of the user are disappeared or the user is not in front of the monitor, then the system stores important data or error messages generated from CVD process (State 8) and replay them when the user returns (State 9). And also in the State 5, if the user is not looking at the right place where the system displays important data or messages, then system leads the user's attention to the desired place (State 7).

The entire gaze detection techniques are hard to be adopted in ready-made programs unless their user-interfaces are reprogrammed. However, the subset of Figure 9, namely, cursor dragging by moving head and button clicking by winking an eye can replace only the role of a ordinary computer mouse, and they are easily adopted in a ready-made program without any changes at already written user-interfaces. These are shown in the Figure 10.

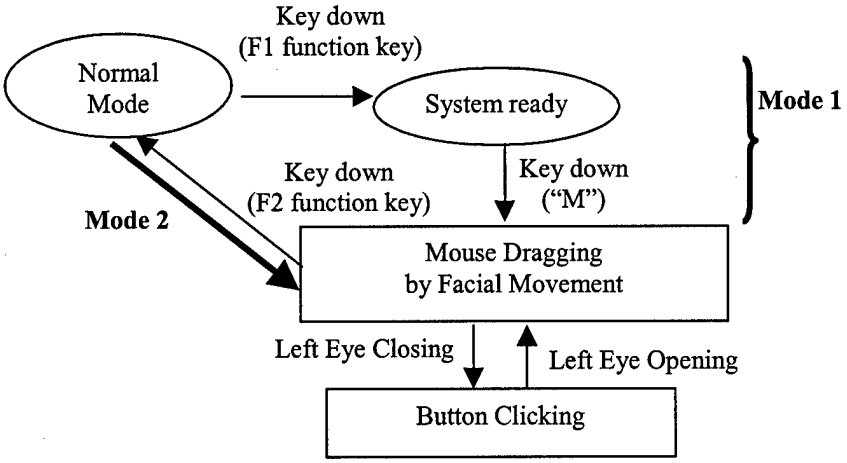


Fig. 10. State Transitions for Ready-made Programs.

5. Evaluation of Gaze Detection Technology in a CVD Control

In this section, experimental results from applying the underlined gaze detection technology to a CVD process control are described. One part of experiments is subjective tests to see how users are feeling when they use the system, and another part is objective tests to measure performances of the system. Tests are performed on 30 students aged from 20 to 30.

5.1 Subjective Tests

Subjective tests are carried out by questionnaires on accuracy, convenience and tiredness.

5.1.1 Accuracy

Practicing Times / Test Items	0 minute	5 minutes	10 minutes
1. Accuracy of Gaze Detection	0.488	0.77	0.863
2. Accuracy of Cursor Dragging	0.738	0.988	1.05

(-2 :Very Inaccurate -1 : Inaccurate 0 : Middle 1 : Accurate 2 :Very Accurate)

Table 4: Subjective Tests on Accuracy

In table 4, test item #1 is on the accuracy of gaze detection felt by users having different practicing times before they actually use the system. The users practiced feel the system more accurate. As the cursor dragging needs only tracing of facial features, it is natural that users think the cursor dragging is more accurate than the gaze detection. Another factor related to the accuracy of cursor dragging is the different pixel sizes of an image (320x240 pixels, in our case) compared to a monitor (1024x768 pixels). Maximum facial movement in usual cases is amount to about 100 pixels in an image, and it corresponds to about 1000 pixels in a monitor. Thus about nine pixels out of 10 pixels on a monitor cannot be accessed by cursor dragging, which leads the users feel the system less accurate. Images with higher resolutions can reduce the inaccuracy of cursor dragging, but require more processing times.

5.1.2 Convenience

Practicing Time / Test Items	0 minute	5 minutes	10 minutes
1. Convenience of Gaze Detection	0.866	1.05	1.3
2. Convenience of Cursor Dragging	0.613	1.113	1.175
3. Convenience of State Transitions	0.613	0.8	0.925

4. Convenience of Command Transfer by Gazing at Same Point for a while(1 sec)	0.863	1.05	1.175
5. Convenience of Command Transfer by Winking Left Eye	0.5	0.8	0.89

(-2 : Very Inconvenient -1 : Inconvenient 0 : Middle 1 : Convenient 2 : Very Convenient)

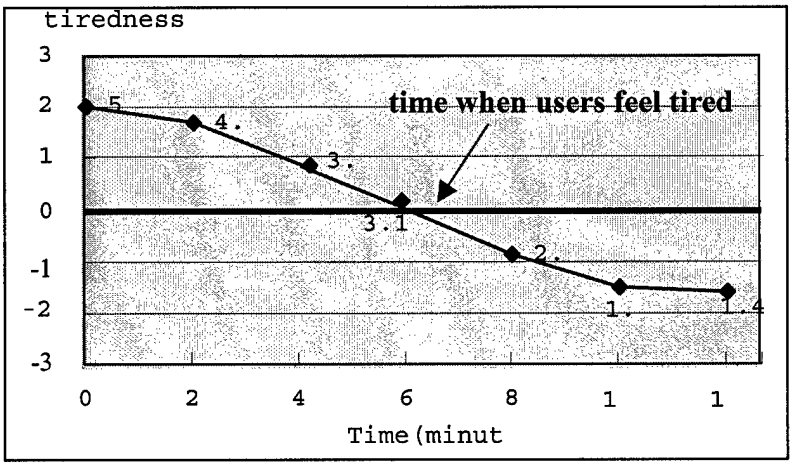
Table 5: Subjective Tests on System Conveniences

From test results, we can see that general users feel the gaze detection is more convenient than the cursor dragging. This is because gazing to place a cursor at a position is accomplished by simply looking at the position, but cursor dragging requires continuous movement of a face or sometimes a whole upper body until the cursor is reached at the desired position. However, the test #2 was made in comparison with the use of ordinary computer mouse. Thus, when the size of a monitor is large or the monitor is somewhat away from the user, usual mouse will be less convenient to use, leading our techniques be relatively convenient.

Test item #4 and #5 are to compare two different ways for giving command signals to the system (button clinking) when the cursor is placed at the desired position. Winking only one eye is not so easy for many users or even impossible for some users. We provide an alternative to it by looking at the same point for more than 1 second. Users feel this is more convenient way. Currently, we are also considering to sending a signal by closing two eyes for a certain time, and opening/closing the mouth.

5.1.3 Tiredness

Tiredness tests are to see how long a user is expected to be in the gaze detection mode without feeling tired. From the testing results in Figure 11, we can see that typical users start to become tired after about 6 minutes. According to that, if we provide ways return to the Normal State or System Ready State(in which the gaze is not detected) without key press as shown in the Fig. 9 and 10 in case gaze detection time is over 6 minutes, users feel less tired than before.



2: not tired, 0: start to be tired, -2: very tired

Fig. 11. Tiredness with Working Times

5.2 Objective Tests

Objective tests are performed on the processing time for the Function 1 described in the Section 4.1, decision errors for the Function 2 and eye blinking.

5.2.1 Job Processing Time

Job processing time is measured on tasks when both hands are busy as the Function 1 described in Section 4.1. Namely, for CVD control the task consists of closing the pipes carrying oxygen (c) and argon (b) by both hands and vaporizer pipe (a) by gaze detection, and then opening the pipes for oxygen and vaporizer by hands and argon by gaze detection. Job processing

time is the average of the time measured three times for the same tasks. They are compared to the cases when ordinary computer mouse is used instead of gaze detection techniques as shown in the Figure 12. We can easily see that controls by ordinary mouse take less time than by gaze detection, but for the more practiced users the time gaps become smaller.

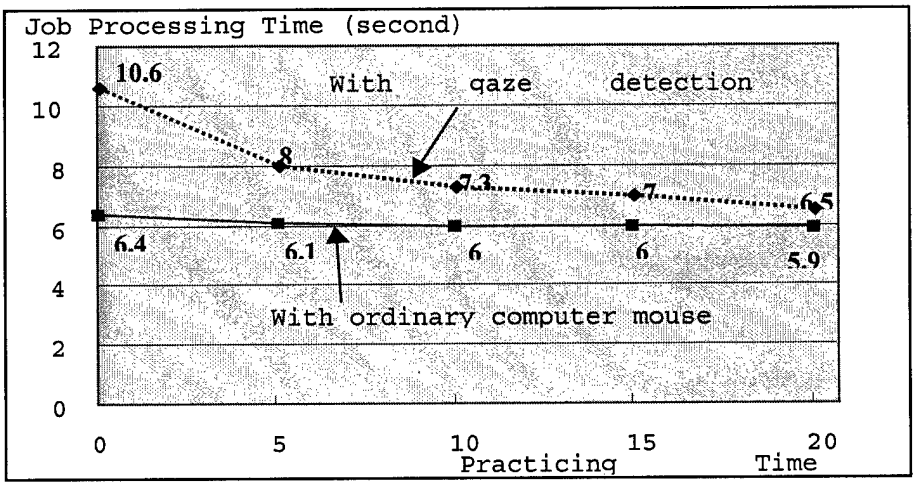


Fig. 12. Comparisons of Job Processing Time with Gaze Detection Techniques to That with Ordinary Computer Mouse

5.2.2 Accuracy on the Decision for the Absence of a User

When the facial features including both eyes, nostrils and lip corners are disappeared, then it means that the user is not in front of the monitor. Experimental results show about Type I error of 8% (misrecognizing user's being absent as being present) and Type II error of 0.1% (misrecognizing user's being present as being absent), where the average error of Type I and II shows minimal value and the errors are due to the backgrounds having similar shapes to the facial features.

5.2.3 Accuracy on the Decision for Blinking of an Eye

Experimental results show Type I error of 6.5% (misrecognizing user's eye winking as no-winking) and Type II error of 0.08% (misrecognizing user's no-winking as winking) of decisions for winking an eye are correct, where the average error of Type I and II shows minimal value. Those users who are not easy to blink only one eye are used to closing both eyes together, which leads decision errors.

6. SUMMARY AND FUTURE WORKS

In this paper, gaze detection techniques for locating the position on a monitor screen that the user is looking at are implemented by the Two Neural Networks Method. One network is for computing the amount of face rotation and another is for face translation. Experimental results show that average RMS error for gaze detection is 1.7" when allowing only face rotation and 3.4" with both face rotation and translation, on a 19" monitor. The proposed method is more practical than the Single Neural Network Method in the sense that when both face rotation and translation are occurred, it performs better. Gaze detection errors can be reduced by improving the methods for locating a face position and facial features, and also by considering the movement of pupils of eyes. We are currently working on these.

For applications of gaze detection to user-interfaces of control programs, we supplement two additional techniques. One is for dragging a cursor by moving face and another is for clicking the cursor by winking one eye. These are to send a command signal to the system with placing the cursor at desired position. All these are applied to a user-interface of a CVD control program in three ways. Firstly, even when both hands are busy in doing a work, the user can still control the monitor screen by gazing, dragging and winking. Secondly, if the user is absent and not in front of the monitor when important states occur, then system can record and replay them later when the user returns. Finally, as the system knows where the user is looking, if an emergency occurs at a different point from the user is looking, then the system can give a warning signal.

Subjective tests on the application of gaze detection techniques to a CVD control show that users feel gaze detection is more convenient but less accurate than cursor dragging, and they generally becomes tired about six minutes after starting to

use the system. Objective tests show that 97% of decisions for the absence of a user and 93.5% of decisions for blinking of an eye are correct, and processing with an ordinary mouse takes less time than with the gaze detection technique, but for the more practiced users the time gaps become smaller.

We are also applying the gaze detection technique to measure the seriousness of a patient paralyzed on head movement and extending to more practical uses in other process controls as well as in monitoring systems.

REFERENCES

1. T.Rikert, M. Jones, 1998. Gaze Estimation using Morphable Models, Proceedings of the 3th International Conference on Automatic Face and Gesture Recognition, Japan, pp. 436-441.
2. A. Azarbayejani, 1993. Visually Controlled Graphics, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 15, No. 6, pp. 602-605.
3. P.Ballard, G.Stockman, 1995. Controlling a Computer via Facial Aspect, IEEE Trans. on System Man and Cybernetics, Vol. 25, No. 4, pp. 669-677.
4. A.Gee, R.Cipolla, 1996. Fast visual tracking by temporal consensus, Image and Vision Computing, Vol. 14, pp. 105-114.
5. J.Heinzmann, A.Zelinsky, 1998. 3-D Facial Pose and Gaze Point Estimation using a Robust Real-Time Tracking Paradigm, Proceedings of the 3th International Conference on Automatic Face and Gesture Recognition, Japan, pp. 142-147.
6. G. R. Park, S. W. Nam, S. C. Han and Jaihie Kim, 1997. Gaze Position Estimation Using Linear Interpolation and 2D Image Informations, the 6th JCEANF, Korea, pp. 255-258.
7. S. W. Nam, G. R. Park, S. C. Han and Jaihie Kim, 1998. A Facial-Feature Tracking Algorithm Based on the Constant Acceleration Model in Multimodal Interface Environment, the 10th Workshop on Image Processing and Understanding, Korea, pp. 209-214.
8. R.C. Gonzalez, R.E. Woods, 1995. Digital Image Processing, Addison-Wesley Publishing Company.